4Ш

УДК 004.89:004.852

В.И. Шинкаренко, Е.Г. Васецкий, Н.Н. Пятковский
Днепропетровский национальный университет железнодорожного транспорта
имени академика В. Лазаряна, г. Днепропетровск, Украина
ccp@diit-70.dp.ua

Потенциальные возможности адаптации алгоритмов

В статье систематизирована информация об адаптации алгоритмов. Сформулирована задача адаптации алгоритмов. Рассмотрены возможные режимы и методы адаптации. Определен минимальный состав адаптивного алгоритма. На основе анализа известных методов и подходов к адаптации алгоритмов в информационных системах, накопленного опыта прикладного программирования выделены признаки полноценного адаптивного алгоритма.

Введение

Одним из основных свойств интеллектуальных систем является способность к адаптации [1], [2]. Присущая растениям, живым организмам, человеку как индивидууму и сообществу людей естественная способность приспосабливаться к окружающей среде распространяется на технические и автоматизированные информационные системы. Наличие у таких систем адаптивных способностей повышает их жизнеспособность, долговечность, удобство использования и другие важные характеристики.

В данной работе рассматриваются возможности адаптации в основном информационных систем. В отличие от технических систем, где адаптация возможна на основании различных технических решений средствами соответствующих устройств и механизмов, адаптация автоматизированных информационных систем может основываться исключительно на адаптационных возможностях алгоритмов.

Несмотря на важность адаптивных свойств алгоритмов в информационных системах, исследование и применение адаптивных алгоритмов в основном ограничивается довольно узким кругом классов алгоритмов. К ним относятся алгоритмы адаптивной обработки сигналов (при передаче по каналам связи) [3-6], обработки изображений [7-10], сжатия данных [11], поиска [12], [13] и некоторые другие.

В фундаментальных работах [12], [14] рассматривалась адаптация разного рода систем с алгоритмической составляющей. В данной работе рассматриваются адаптивные возможности самих алгоритмов. Алгоритм сам по себе представляет собой систему, обладающую множеством особенностей, которые отличают ее от технических, социальных и других систем.

Цель данной статьи: на основе анализа известных методов и подходов к адаптации алгоритмов в информационных системах, накопленного опыта прикладного программирования выделить существенные признаки, позволяющие идентифицировать алгоритм как адаптивный, классифицировать алгоритмы по этим признакам. Это позволит уточнить термин «адаптация алгоритма», снизить уровень интуитивной составляющей в терминологии и, как следствие, совершенствовать процессы разработки адаптивных алгоритмов, способствовать систематическому подходу к интеллектуализации информационных систем средствами адаптации.

Постановка задачи

Ввиду неустановившейся терминологии исследователи довольно часто называют адаптивными алгоритмы, в которых отсутствуют признаки адаптации. В то же время действительно адаптивные алгоритмы таковыми не считают. И для того и другого есть определенные основания.

Рассмотрим пример простого алгоритма решения квадратного уравнения (рис. 1).

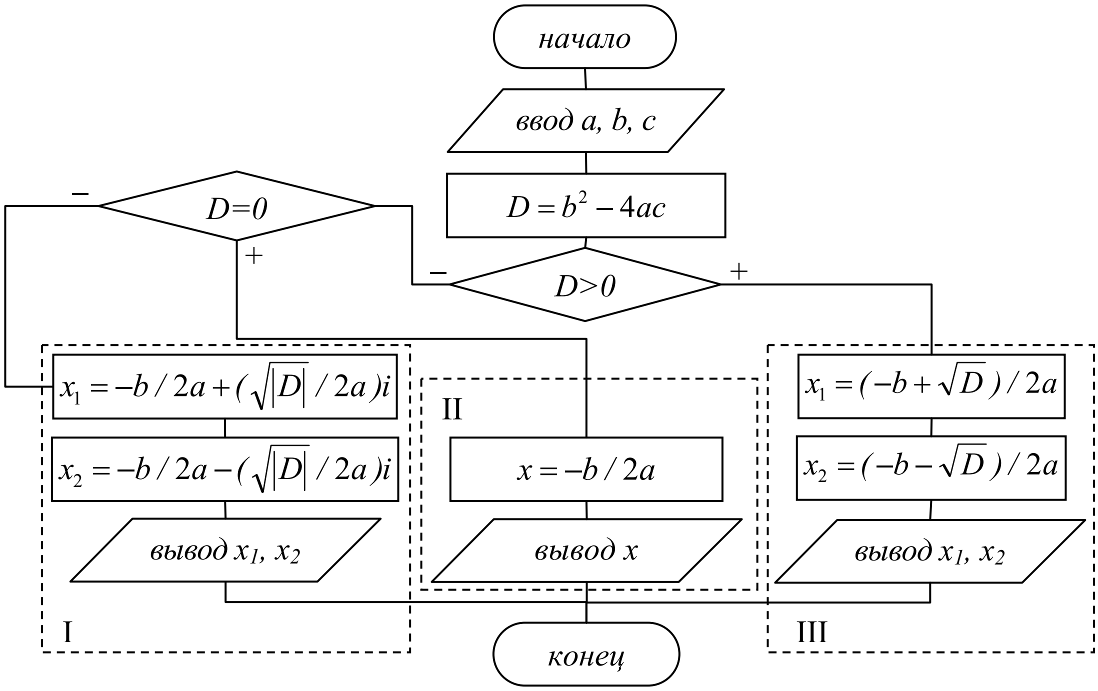


Рисунок 1 – Алгоритм решения квадратного уравнения

В одних случаях выполнения выполняется блок инструкций I, в других – II или III. Можно ли на этом основании считать алгоритм адаптивным (приспосабливающимся к данным)?

С одной стороны, поведение алгоритма изменяется в зависимости от данных. При различных коэффициентах уравнения алгоритм функционирует по-разному и выдает принципиально разные значения. Более того, изменяется и его временная эффективность. С другой стороны, основанием к таким выводам является всего лишь наличие в алгоритме такой управляющей конструкции, как ветвление, которая присутствует практически во всех алгоритмах. Тогда, согласно вышесказанному, практически каждый алгоритм можно считать адаптивным.

Основной задачей данной работы является выделение тех признаков или свойств алгоритмов, которые дают основания считать его адаптивным.

Уточнение задачи адаптации

Применительно к алгоритмам определение Ахоффа [15] – «адаптивность это способность индивида или системы модифицировать себя или свое окружение, когда происходит неблагоприятное для них изменение для того, чтобы хотя бы частично компенсировать потерю эффективности» – можно сформулировать следующим образом. Адаптивным будет алгоритм, который модифицируется в аспекте представления и/или выполнения при неблагоприятном окружении, «чтобы хотя бы частично компенсировать потерю эффективности».

Надо понимать, что минимальные требования компенсации эффективности не противоречат возможности повышения эффективности при благоприятных условиях функционирования алгоритмов.

В биологических, технических и других системах индивид или объект существует и функционирует в одном, постепенно или кардинально изменяющемся, окружении. Алгоритм же может легко дублироваться и одновременно функционировать в различном окружении. Это уже служит основанием для адаптации. Не всегда оправданно говорить о неблагоприятном окружении, неблагоприятным может быть уже само разнообразие окружения.

Изменение алгоритмом окружающей среды рассматривать не будем по причине того, что во внешнем окружении алгоритм способен поменять только программную среду выполнения (запустить или остановить другие процессы-алгоритмы, изменить их приоритеты); это может отразиться только на временной эффективности алгоритма и в этом случае целесообразно рассматривать не отдельный алгоритм, а систему алгоритмов.

Перейдем к формальной постановке задачи адаптации.

Обозначим $A|_X^Y$ – алгоритм с областью определения X и областью значения Y [16].

Пусть дано некоторое конечное или бесконечное множество алгоритмов $A_i|_{X_i}^{Y_i}$, каждый из которых реализует некоторую функцию $y=\varphi(x)$, где $x \in X$, $y \in Y$. Алгоритм A_i имеет некоторое множество параметров U_i, которое оказывает влияние на эксплуатационные характеристики алгоритма и, в общем случае, на выходные данные Y_i, т.е. $A_i|_{X_i,U_i}^{Y_i}$.

В зависимости от множества параметров среды функционирования $g_k \in G$ при конкретном выполнении алгоритма $A_i|_{X_i,U_i}^{Y_i}$ проявляются его характеристики качества с показателями $\rho_m(A_i|_{X_i,U_i}^{Y_i}, G)$.

К эксплуатационным характеристикам качества алгоритмов относятся временная и функциональная эффективность [17], точность, надежность (хотя вопросы надежности алгоритмов практически не изучены), ресурсоемкость (размер и качество ресурсов: оперативной памяти, памяти на внешних запоминающих устройствах, каналов связи и др.).

Тогда задача адаптации заключается в следующем: для различных условий функционирования $\overline{G}_j \in \overline{G}$ необходимо найти алгоритм $A_i^*|_{X_i,U_i^*}^{Y_i}$ и его параметры U_i^*, удовлетворяющие системе отношений:

$$\forall \overline{G}_j \in \overline{G} : \rho_m(A_i^*|_{X_i,U_i^*}^{Y_i}, G_j) \le \rho_m(A^b|_{X\ ,U^b}^{Y}, G_j), \quad (1)$$

и для некоторых $\overline{G}_j$ неравенство строгое. Здесь A^b, U^b – любые априорно выбранные алгоритм и управляющие параметры соответственно. Адаптивный алгоритм должен быть в различных условиях функционирования не хуже любого известного.

Согласно (1) неудачные изменения алгоритма при неблагоприятном изменении внешней среды нельзя считать адаптацией.

Адаптация, оптимизация и компенсация

Сформулированная задача близка по математической постановке к задаче оптимизации [12] и может быть построена соответствующая функция цели. Возникает естественный вопрос. Почему адаптация, а не оптимизация?

Особенность изучаемого объекта (рис. 2) заключается в сложности и практически невозможности построения его модели.

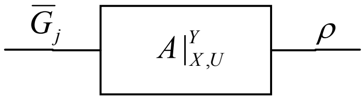


Рисунок 2 – Алгоритм как объект адаптации

Например, временная эффективность алгоритма (ρ) зависит от объема, типа, значений входных данных, а также программной среды создания и функционирования выполняемого файла и архитектуры ЭВМ ($\overline{G}_j$) [17]. С учетом сложных и разных аппаратно реализованных алгоритмов кэширования данных, конвейеризации, предконвейерной оптимизации и множества других существенных факторов архитектуры ЭВМ на основе процессоров, модель выполнения алгоритма построить чрезвычайно сложно и к тому же ее необходимо постоянно модифицировать.

Так как задача оптимизации формулируется как нахождение экстремума по известной модели объекта или процесса [12], сформулировать и решить задачи оптимизации алгоритмов и задачи оптимального управления алгоритмами необычайно сложно. Это возможно лишь в отдельных случаях. Например, задачи нахождения оптимального по точности и оптимального по сложности алгоритма [18] решаются без учета особенностей исполнительных устройств.

В [12] задача адаптации сложных систем сводится к задаче определения экстремума. В силу вышесказанного требованиям адаптации больше подходит задача с ограничениями (1). Понятия «лучшего» алгоритма во многих случаях не существует. Алгоритм сортировки для одних данных (например, уже отсортированных) может оказаться лучше, чем другой алгоритм, а для других данных – хуже. А формулировать оптимизационную задачу для каждого набора данных бессмысленно.

Ограничения задачи адаптации (1) можно несколько ослабить:

$$\forall \overline{G}_j \in \overline{G} : \rho_m(A_i^* |_{X_i,U_i^*}^{Y_i}, G_j) \le \rho_m^0, \qquad (2)$$

где ρ_m^0 – необходимый уровень показателя качества, который должен обеспечивать адаптивный алгоритм. Если ρ_m – показатели времени выполнения алгоритма, требуемой памяти и точности вычислений, то (2) определяет задачу адаптации как требование Т-эффективности алгоритмов [19].

Если рассматривать не многообразие внешней среды при эксплуатации алгоритмов, а лишь неблагоприятные изменения внешней среды, то задачу адаптации можно рассматривать как известную задачу компенсации [12]. Падение эффективности алгоритмов при дрейфе параметров внешней среды компенсируется адаптацией.

Адаптация и обучение

В настоящее время очень популярными являются нейронные сети и соответствующие алгоритмы. Возникает вопрос: насколько такие алгоритмы являются или могут быть адаптивными? Вопрос связан с тем, что информационные системы, основанные на нейросетях, обладают свойством обучаемости, а обучаемые системы, как и адаптивные, могут компенсировать неблагоприятное изменение окружения.

Здесь надо учитывать представление нейросети. Если это нагруженный ориентированный граф, то он является входными данными для алгоритма обработки (например,

распознавания образов). Эти данные являются необходимыми данными для реализации функциональности и не являются управляющими. Поэтому классическое представление нейросетевых алгоритмов не является адаптивным.

В случаях непрерывного обучения [20], представления нейронов в виде объектов (в терминах объектно-ориентированного программирования) при реорганизации нейросетей, других адаптирующих преобразований, нейросетевые алгоритмы, как и любые другие, могут приводиться к адаптивным.

Внешняя среда для алгоритмов

Естественно внешними для алгоритмов являются входные данные. Данные могут поступать на вход алгоритмов с внешних устройств: клавиатуры, накопителя на магнитных дисках, модемов и др. Непосредственно доступ к данным обеспечивается операционной системой и возможно прикладным программным обеспечением.

В случае многократного выполнения алгоритмов в стабильной окружающей среде можно решать задачу адаптации алгоритмов к потокам данных. В данном случае под потоком данных следует понимать множество наборов входных данных, каждый такой набор включает необходимые и достаточные данные для однократного выполнения алгоритма. В этом случае для успешной адаптации статистические характеристики наборов входных данных не должны изменяться или изменяться незначительно. Также должны быть стабильными все остальные факторы окружающей среды.

Внешней средой алгоритма является программно-аппаратная среда его функционирования. Программная среда состоит из операционной системы, прикладных программ, одновременно выполняемых с адаптируемым алгоритмом, а также вспомогательные прикладные и системные функции (например, библиотеки математических функций, API функций).

Аппаратная среда характеризуется доступными для алгоритма ресурсами, такими, как процессор, оперативная память и др.

И еще один фактор внешней среды – критерии качества алгоритма, которые тоже могут изменяться в процессе его эксплуатации. Так, например, для алгоритмов сжатия данных существенными являются степень сжатия данных, время архивации и время разархивации. На рис. 3 приведены элементы управления критериями (для трех критериев).

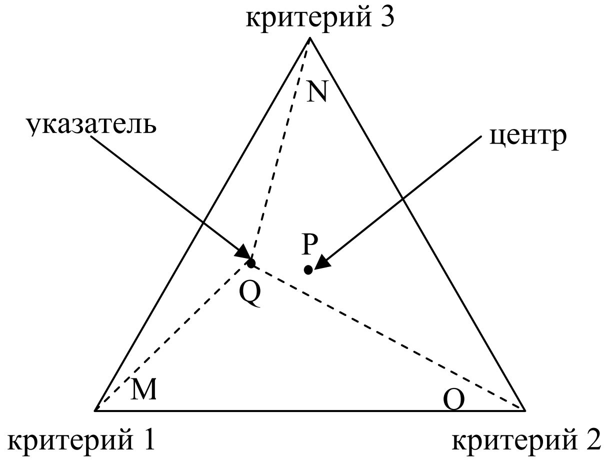


Рисунок 3 – Элементы интерфейса пользователя для управления критериями

Согласно заданным пользователям соотношениям критериев может формироваться компромиссный критерий (для трех критериев):

$$\rho = (1 - S_{MQ} / S_{MP}) a_1 \rho_1 + (1 - S_{OQ} / S_{OP}) a_2 \rho_2 + (1 - S_{NQ} / S_{NP}) a_3 \rho_3 ,$$

где S_{MP} – расстояние между точками M и P, a_i – нормирующие коэффициенты.

Состав и функции адаптивного алгоритма

Пусть имеется «обычный» неадаптивный алгоритм. Изначально он выполняет некоторое функциональное назначение и обладает набором характеристик качества. Для придания алгоритму способности к адаптации необходима доработка алгоритма в виде адаптирующего алгоритма. Начальный алгоритм естественно назвать адаптируемым.

Адаптирующий алгоритм может быть выполнен как отдельный алгоритм и отдельно от адаптируемого выполняться. С другой стороны, он может быть внедрен в адаптируемый алгоритм в виде его дополнительной части или частей.

Функциональное назначение адаптирующего алгоритма – повышение одного или нескольких показателей качества адаптируемого алгоритма при неблагоприятных изменениях внешней среды, либо при эксплуатации адаптируемого алгоритма в существенно различных средах.

Причинами или основанием для выполнения адаптации адаптирующим алгоритмом являются изменения внешней среды или опосредствованно снижение эффективности адаптируемого алгоритма. Для отслеживания таких изменений адаптирующий алгоритм должен содержать (как часть или отдельно) алгоритм измерений.

Адаптация возможна на основе измерения:

– каких-то показателей самих данных. Например, яркости изображения [21];

– статистических показателей входных данных или потоков входных данных;

– состояния аппаратного окружения (размера доступной оперативной памяти, количества ядер процессора и т.п.);

– состояния программного окружения (наличия и потребности в ресурсах параллельно выполняемых процессов, возможностей операционной системы и др.);

– текущих показателей качества адаптируемого алгоритма.

Адаптирующий алгоритм (с алгоритмом измерений) может быть довольно сложным [22] или предельно простым (несколько управляющих конструкций [21]).

Для выполнения адаптивного воздействия необходима система знаний о том, что нужно для достижения положительного эффекта адаптации. Такие знания могут быть представлены в виде базы знаний [23], и тогда адаптирующий алгоритм должен иметь соответствующую функциональность работы с базой знаний либо знаниями, заложенными непосредственно в адаптирующий алгоритм (при принятии решений управляющими конструкциями).

Методы адаптации алгоритмов

Различают структурную, альтернативную и параметрическую адаптацию [12] сложных систем. Применительно к алгоритмам можно говорить также о структурно-параметрической и параметро-структурной адаптации.

Процесс *параметрической адаптации алгоритмов* заключается в определении управляющих параметров [8], [11], [13], [20], [24], [25], удовлетворяющих условию (1). Во время эксплуатации сам алгоритм, его структура не изменяются.

Входные данные алгоритма можно разделить на две группы: данные, необходимые для обеспечения функциональности, и параметры для управления качеством.

Параметрически адаптивный алгоритм можно представить как $A|_{X,U}^{Y}$.

Структурная адаптация алгоритма подразумевает изменение структуры алгоритма в процессе его выполнения или эксплуатации (многократного выполнения) [12], [13], [17]. Эффект адаптации достигается применением управляющих воздействий на структуру алгоритма $A(U)|_{X}^{Y}$.

Частным случаем структурной адаптации является *альтернативная адаптация*. Изменение структуры алгоритма при альтернативной адаптации состоит в полной замене алгоритма одним из набора функционально эквивалентных алгоритмов [23], [24].

В качестве инструментария для реализации структурной адаптации алгоритмов могут быть использованы трансляторы и интерпретаторы.

Интерпретаторы допускают изменения текста программы (алгоритма, представленного средствами языка программирования) в процессе выполнения. Программа может сама себя модифицировать, изменять свою структуру. Это устраняет все проблемы с изменением структуры программы в процессе выполнения. Такой подход имеет единственный существенный недостаток, связанный с самой концепцией интерпретации. Так как каждая инструкция (оператор) перед выполнением транслируется (переводится в исполнимый код) это существенно влияет на временные характеристики алгоритмов. Поэтому, если целью адаптации является повышение временной эффективности алгоритмов, то такой подход не оправдывает себя.

Использование трансляторов в качестве инструментария структурной адаптации предполагает при изменении алгоритма (его структуры) выполнять его повторную трансляцию.

Разработка структурно-адаптивных алгоритмов – достаточно трудоемкий процесс. Такие алгоритмы – редкость в научных исследованиях и тем более в практическом программировании. Однако, как показывают результаты, эффективность таких алгоритмов может быть повышена в десятки раз [22], [26].

Авторами предложен подход к структурной адаптации алгоритмов на метаалгоритмической основе [22], разработан соответствующий инструментарий [27], а также разработаны структурно-адаптивные алгоритмы сортировки [22] и сжатия данных [26].

Текст адаптируемого алгоритма синтезируется на основе так называемого метаалгоритма и рекомендаций системы анализа, транслируется и под управлением измерительной системы выполняется. Такая последовательность продолжается до тех пор, пока стабильно будут синтезироваться одинаковые алгоритмы.

Метаалгоритм – специальным образом заданный алгоритм, на основе которого могут быть построены конкретные алгоритмы, обобщенный алгоритм решения некоторой задачи. Формирование метаалгоритма выполняется на основе модифицированного метода пошаговой детализации. Суть и особенность модификации заключаются в том, что на каждом шаге детализации абстрактный оператор может иметь несколько реализаций [22].

Одним из способов реализации структурной адаптации является *параметро-структурная адаптация*. При структурной адаптации изменение структуры алгоритма выполняется в аспекте представления. Выполняется замена одних частей алгоритма другими (рис. 4а). Ту же замену можно делать во время выполнения. В зависимости от некоторого управляющего параметра-признака может выполняться одна или другая часть алгоритма (рис. 4б). Фактически структура алгоритма не изменяется, но сохраняется эквивалентность по выполнению со структурно адаптивным алгоритмом.

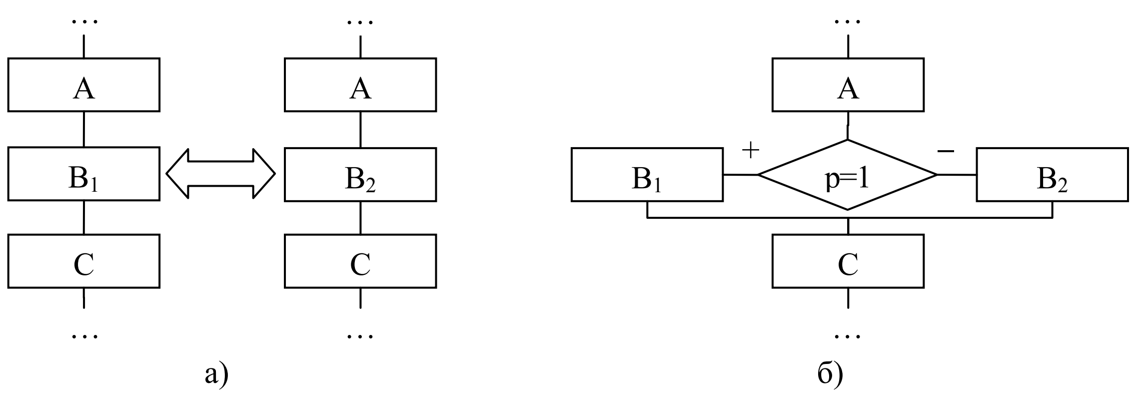


Рисунок 4 – Структурная (а) и параметро-структурная адаптация (б)

Параметро-структурная адаптация наиболее простая в реализации, но адаптивный алгоритм уступает структурно-адаптивному по временным характеристикам и объему требуемой памяти.

Структурно-параметрическая адаптация объединяет возможности структурной и параметрической адаптации. Управление адаптацией состоит в изменении структуры алгоритма и одновременно параметров управления качеством $A(U)\,|_{X,U}^{Y}$.

Режимы адаптации

На вопрос «когда, в какое время выполняется процесс адаптации алгоритма?» существует два принципиально разных подхода.

Во-первых, процесс адаптации может быть частью процесса однократного выполнения алгоритма. При этом адаптирующий алгоритм является частью адаптируемого алгоритма. Такую адаптацию называют *online* адаптацией [14], или *СНС* (сегодня на сегодня) адаптацией [23].

Во-вторых, адаптация может выполняться до или между выполнениями адаптивного алгоритма. В этом случае адаптацию называют *offline* адаптацией [14], или *СНЗ* (сегодня на завтра) адаптацией [23].

Адаптирующий алгоритм может выполняться заблаговременно программистом. Тогда говорят о статической адаптации алгоритма [14], [28], [29], что точнее называть специализацией алгоритма. Не стоит смешивать специализацию с адаптацией.

Динамическая *СНЗ* адаптация предполагает периодическое выполнение адаптирующего алгоритма между выполнениями адаптируемого. Организационно это может быть реализовано следующим образом.

Адаптивный алгоритм эксплуатируется на одних технических средствах, а адаптирующая система – на аналогичных других. Для адаптации надо обеспечить идентичность внешней среды при эксплуатации и адаптации. В качестве технических средств адаптации могут использоваться средства, находящиеся в холодном резерве, либо те же средства, на которых эксплуатируется адаптивный алгоритм, но во время их простоя. Многие вычислительные системы неравномерно загружают технические средства. Например, системы, связанные с торговлей, учебным процессом, характеризуются значительными простоями технических средств в ночное время. Эти ресурсы могут без дополнительных затрат быть использованы для адаптации.

Для достижения цели *СНЗ* адаптации необходима стабильность среды выполнения адаптируемого алгоритма. В каких-то случаях стабильность среды должна контролироваться в процессе эксплуатации, а в каких-то – достаточно аргументированного заключения на основе анализа особенностей задачи и целей адаптации.

Выделяют также *превентивную* и *реактивную* адаптацию [28]. *Превентивная* адаптация применяется заблаговременно, до неблагоприятных изменений окружающей среды или до эксплуатации алгоритма в неблагоприятной среде. *Реактивная* адаптация является реакцией на неблагоприятные изменения и выполняется, возможно, с некоторым запаздыванием.

Выводы

На основании выполненного анализа установлено, что полноценный адаптивный алгоритм имеет все признаки адаптации:

– должны быть предъявлены требования к качеству алгоритма;

– можно выделить элементы внешней среды, к которым выполняется адаптация;

– должен присутствовать адаптирующий алгоритм, как отдельный алгоритм или как часть адаптируемого;

– адаптация должна выполняться либо в процессе выполнения алгоритма, либо в процессе его эксплуатации в автоматическом режиме;

– для управления качеством необходимо наличие измерительного алгоритма (или части адаптируемого алгоритма), который должен отслеживать состояние внешней среды и/или качество адаптируемого алгоритма;

– можно идентифицировать метод адаптации как структурную, альтернативную, параметрическую, структурно-параметрическую или параметро-структурную адаптацию;

– адаптация позволяет повысить эффективность алгоритма при выполнении или эксплуатации в разнородной или изменяющейся внешней среде (положительный эффект достигается).

Таким образом, адаптивный алгоритм должен иметь адаптивную, адаптирующую и измерительную составляющую в виде составной части или отдельных алгоритмов.

Алгоритм, имеющий только часть из этих признаков, можно считать адаптивным лишь условно.

Литература

1. Босов А.А. Понятие «искусственный интеллект» и перспективы создания реального искусственного интеллекта / А.А. Босов, В.И. Шинкаренко, В.В. Скалозуб // Искусственный интеллект : материалы Междунар. науч.-техн. конф., (31 сент. 2002) . – Таганрог, 2002. – С. 74-78.
2. Мейтус В.Ю. Программирование и интеллект / В.Ю. Мейтус // Проблеми програмування. – 2009. – № 1. – С. 17-27.
3. Титаренко Л.А. Адаптивная пространственная обработка сигналов в условиях априорной неопределенности / Л.А. Титаренко. – Х. : ХНУРЭ; Коллегиум, 2004.– 216 с.
4. Адаптивные фильтры / [П.М. Грант, К.Ф.Н. Коуэн, Б. Фрилендер, Д.Р. Тернер и др.] ; под ред. К.Ф.Н. Коуэна и П.М. Гранта. – М. : Мир, 1988. – 392 с.
5. Vijaykumar V.R. Modified Adaptive Filtering Algorithm for Noise Cancellation in Speech Signals / V.R. Vijaykumar, P.T. Vanathi, P. Kanagasabapathy // Electronics and Electrical Engineering. – 2007. – № 2(74). – P. 17-20.
6. Джиган В.И. Многообразие алгоритмов адаптивной фильтрации по критерию наименьших квадратов / Джиган Виктор Иванович // Современная электроника. – 2008. – № 3. – С. 32-39.
7. Егоров А.А. Метод адаптивного контрастирования мультиспектральных изображений для повышения достоверности нечеткой кластеризации / А.А. Егоров // Искусственный интеллект. – 2008. – № 3. – С. 306-314.
8. Louban R. Image processing of edge and surface defects / R. Louban. – Berlin : Springer Series in Materials Science, 2009. – 168 p.
9. Роенко А.А. Применение устойчивых оценок параметров выборок данных при обработке изображений / А.А. Роенко, Д.В. Февралев, Н.Н. Пономаренко, В.В. Лукин // Восточно-Европейский журнал передовых технологий. – 2007. – № 3/3(27). – С. 10-15.
10. Крашенников В.Р. Основы теории обработки изображений / Крашенников В.Р. – Ульяновск : УлГТУ, 2003. – 70 с.

11. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / [Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин]. – М. : Диалог-МИФИ, 2003. – 384 с.
12. Растригин Л.А. Адаптация сложных систем / Л.А. Растригин. – Рига : Зинатне, 1981. – 375 с.
13. Курейчик В.М. Поисковая адаптация: теория и практика / Курейчик В.М., Лебедев Б.К., Лебедев О.К. – Москва : Физматлит, 2006. – 272 с.
14. Теслер Г.С. Новая кибернетика / Г.С. Теслер. - К. : Логос, 2004. - 404 с..
15. Акофф Р. О целеустремленных системах / Р. Акофф, Ф. Эмери. – М. : Советское радио, 1974. – 272 с.
16. Шинкаренко В.И. Структурные модели алгоритмов в задачах прикладного программирования. Часть I : формальные алгоритмические структуры / В.И. Шинкаренко, В.М. Ильман, В.В. Скалозуб // Кибернетика и системный анализ. – 2009. – № 3. – С. 3-14.
17. Шинкаренко В.И. Экспериментальные исследования алгоритмов в программно-аппаратных средах : монография / В.И. Шинкаренко. – Д. : Изд-во Днепропетр. нац. ун-та ж.-д. трансп. им. акад. В. Лазаряна, 2009. – 279 с.
18. Трауб Дж. Общая теория оптимальных алгоритмов / Дж. Трауб, Х. Вожьняковский. – М. : Мир, 1983.– 382 с.
19. Бабич М.Д. Вычислительный эксперимент в проблеме оптимизации вычислений / М.Д. Бабич, В.К. Задирака, И.В. Сергиенко // Кибернетика и системный анализ. – 1999. – № 1, ч. 1. – С. 51-62; № 2, ч. II. – С. 59-79.
20. Хайкин С. Нейронные сети. Полный курс / С. Хайкин. – М., СПб., К. : Вильямс, 2008.– 1103 с.
21. Ульшин В.А. Адаптивный алгоритм считывания перфорированного кода / В.А. Ульшин, М.В. Дубровкина // Искусственный интеллект. – 2007. – № 3. – С. 113-121.
22. Шинкаренко В.И. Структурная адаптация алгоритмов на основе полиморфизма / В.И. Шинкаренко // Математичні машини та системи. – 2009. – № 2. – С. 28-44.
23. Шинкаренко В.И. Знание-ориентированный подход к адаптации алгоритмов / В.И. Шинкаренко // Искусственный интеллект. – 2008. – № 3. – С. 388-397.
24. Estivill-Castro V. A survey of adaptive sorting algorithms / Vladimir Estvill-Castro, Derrick Wood // ACM Computing Surveys. – 1992. – Volume 24 Issue 4. – P. 441-476.
25. Herrera F. Adaptation of genetic Algorithm parameters based on fuzzy logic controllers / Francisco Herrera, Manuel Lozano // Genetic Algorithms and Soft Computing. – 1996. - № 8. – P. 95-125.
26. Шинкаренко В.И. Структурная адаптация алгоритмов сжатия данных на метаалгоритмической основе / В.И. Шинкаренко, Г.Г. Кроль, Е.Г. Васецкий, Т.Н. Мажара // Искусственный интеллект. – 2009. – № 4. – С. 104-111.
27. Шинкаренко В.И. Методы и средства структурной адаптации алгоритмов на метаалгоритмической основе / В.И. Шинкаренко, Г.Г. Кроль, И.В. Литвин, Е.Г. Васецкий // Искусственный интеллект. – 2009. – № 3. – С. 105-113.
28. Дерецький В. Підхід до адаптивної композиції сервісів в Semantic WEB середовищі на основі потоків робіт програмних агентів / В. Дерецький // Проблеми програмування. – 2008. – № 2-3. – С. 403-430.
29. Цейтлин Г.Е. Введение в алгоритмику / Г.Е. Цейтлин. – Киев : Сфера, 1998. – 310 с.

Literatura

1. Bosov A.A., Shinkarenko V.I., Skalozub V.V. Ponjatie «iskusstvennyj intellekt» i perspektivy sozdanija real'nogo iskusstvennogo intellekta. – 2002.
2. Mejtus V.Ju. Programmirovanie i intellekt. – 2009.
3. Titarenko L.A. Adaptivnaja prostranstvennaja obrabotka signalov v uslovijah apriornoj neopredelennosti. – 2004.
4. Grant P.M., Koujen K.F.N. Frilender B, Terner D.R. Adaptivnye fil'try. – 1988.
5. Vijaykumar R., Vanathi P. T., Kanagasabapathy P. Modified Adaptive Filtering Algorithm for Noise Cancellation in Speech Signals. – 2007.
6. Dzhigan V.I. Mnogoobrazie algoritmov adaptivnoj fil'tracii po kriteriju naimen'shih kvadratov. – 2008.
7. Egorov A.A. Metod adaptivnogo kontrastirovanija mul'tispektral'nyh izobrazhenij dlja povyshenija dostovernosti nechetkoj klasterizacii. – 2008.
8. Louban R. Image processing of edge and surface defects. – 2009.
9. Roenko A.A., Fevralev D.V., Ponomarenko N.N., Lukin V.V. Primenenie ustojchivyh ocenok parametrov vyborok dannyh pri obrabotke izobrazhenij. – 2007.
10. Krashennikov V.R. Osnovy teorii obrabotki izobrazhenij. – 2003.
11. Vatolin D., Ratushnjak A., Smirnov M., Jukin V. Metody szhatija dannyh. Ustrojstvo arhivatorov, szhatie izobrazhenij i video. – 2003.

12. Rastrigin L.A. Adaptacija slozhnyh system. – 1981.
13. Kurejchik V.M., Lebedev B.K., Lebedev O.K. Poiskovaja adaptacija: teorija i praktika. – 2006.
14. Tesler G.S. Novaja kibernetika. – 2004.
15. Akoff R., Jemeri F. O celeustremlennyh sistemah. – 1974.
16. Shinkarenko V.I., Il'man V.M., Skalozub V.V. Strukturnye modeli algoritmov v zadachah prikladnogo programmirovanija. – 2009.
17. Shinkarenko V.I. Jeksperimental'nye issledovanija algoritmov v programmno-apparatnyh sredah. – 2009.
18. Traub Dzh., Vozh'njakovskij H. Obshaja teorija optimal'nyh algoritmov. – 1983.
19. Babich M.D., Zadiraka V.K., Sergienko I.V. Vychislitel'nyj jeksperiment v probleme optimizacii vychislenij. – 1999.
20. Hajkin S. Nejronnye seti. Polnyj kurs. – 2008.
21. Ul'shin V.A., Dubrovkina M.V. Adaptivnyj algoritm schityvanija perforirovannogo koda. – 2007.
22. Shinkarenko V.I. Strukturnaja adaptacija algoritmov na osnove polimorfizma. – 2009.
23. Shinkarenko V.I. Znanie-orientirovannyj podhod k adaptacii algoritmov. – 2008.
24. Estivill-Castro V., Wood D. A survey of adaptive sorting algorithms. – 1992.
25. Herrera F., Lozano M. Adaptation of genetic Algorithm parameters based on fuzzy logic controllers. – 1996.
26. Shinkarenko V.I., Krol' G.G., Vaseckij E.G., Mazhara T.N. Strukturnaja adaptacija algoritmov szhatija dannyh na metaalgoritmicheskoj osnove. – 2009.
27. Shinkarenko V.I., Krol' G.G., Litvin I.V., Vaseckij E.G. Metody i sredstva strukturnoj adaptacii algoritmov na metaalgoritmicheskoj osnove. – 2009.
28. Derec'kij V. Pidhid do adaptivnoї kompoziciї servisiv v Semantic WEB seredoviwi na osnovi potokiv robit programnih agentiv. – 2008.
29. Cejtlin G.E. Vvedenie v algoritmiku. – 1998.

В.І. Шинкаренко, Є.Г. Васецький, М.М. Пятковський

Потенційні можливості адаптації алгоритмів

У статті систематизована інформація щодо адаптації алгоритмів. Сформульована задача адаптації. Розглянуті можливі режими та методи адаптації. Визначено мінімальний склад адаптивного алгоритму. На основі відомих методів та підходів до адаптації алгоритмів у інформаційних системах, накопиченого досвіду прикладного програмування визначені ознаки повноцінного адаптивного алгоритму.

V.I. Shynkarenko, E.G. Vasetsky, N.N. Pyatkovskiy

The Hotential for Algorithm Adaptation

This article systematizes information about algorithm adaptation. The problem of algorithm adaptation is defined. Possible modes and methods of adaptation are discussed. Minimal composition of adaptive algorithm is defined. Attributes of complete adaptive algorithm are identified based on analysis of known methods and approaches to algorithm adaptation in information systems and experience in applied programming.

Статья поступила в редакцию 08.07.2011.